NASA Contractor Report 177628

IN-67
193169
17 P

Achieving High Data Reduction with Integral Cubic B-Splines

Jin J. Chou

(NASA-CR-177628) ACHIEVING HIGH DATA REDUCTION WITH INTEGRAL CUBIC B-SPLINES (Computer Sciences Corp.) 17 p

N94-21881

Unclas

G3/67 0198169

CONTRACT NAS2-12961
October 1993

National Aeronautics and
Space Administration

NASA Contractor Report 177628

Achieving High Data Reduction with Integral Cubic B-Splines

Jin J. Chou

Computer Sciences Corporation
P.O. Box 390757
Mountain View, CA 94039

Prepared for
Ames Research Center
CONTRACT NAS2-12961
October 1993

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

Achieving High Data Reduction with Integral Cubic B-Splines

Jin J. Chou [†]
Computer Sciences Corporation
P.O. Box 390757
Mountain View, CA 94039

Abstract

During geometry processing, tangent directions at the data points are frequently readily available from the computation process that generates the points. It is desirable to utilize this information to improve the accuracy of curve fitting and to improve data reduction.

This paper presents a curve fitting method which utilizes both position and tangent direction data. This method produces G^1 nonrational B-spline curves. From the examples, the method demonstrates very good data reduction rates while maintaining high accuracy in both position and tangent direction.

1 Introduction

Path tracing is a common technique utilized by many numerical processes in computers, e.g., computing the intersection between surfaces, evaluating the silhouette curves, and solving partial differential equations. Oversampling is a method frequently used by tracing algorithms to overcome uncertainty in tracing step selection and to ease various stability problems. A large amount of

[*]This work was performed for NASA under Contract NAS 2-12961
[†]To be submitted as a NASA Contractor Report

points results from tracing, partially due to oversampling. For the convenience of subsequent data handling and storage, it is desirable to fit the data by curves with less data. Hence data reduction is an important aspect of fitting traced data. In this paper, the word fitting means to pass a curve close to the data but not necessarily exactly through the data.

Tangent direction at the data points is usually used to determine the marching direction in path tracing; hence it is the by-product of the tracing algorithms. Traditional curve fitting algorithms seldom take advantage of this tangent information to improve the accuracy of curve fitting or to enhance data reduction.

Robust tracing algorithms need to detect discontinuity in the path curve. Hence, the resulting points can be grouped into pieces forming continuous paths. Thus, in this paper we assume the given data set forms a continuous path.

The problem at hand is different than fitting inaccurate data, that is data with relatively large error, e.g., digitized data. In such cases, the fitting curve is sought to best approximate the shape, not the individual points. High order information such as tangent is meaningless in such case. However, in tracing, the data, both position and tangent direction, are usually considered to be "exact," and the fitting curve is required to be as close to the data as possible.

Cubic spline fits or least-squares spline fits have been traditionally used for fitting B-spline curves [1, 2, 3, 4, 5, 10]. Cubic spline fits always produce more data than the input. To achieve a given tolerance with the least-squares methods, a fit-then-test procedure with an increasing number of control points is necessary. When the number of control points is increased to the number of input data, least-squares methods converge to interpolation. According to the author's experience, least-squares methods produce reasonable results when the required fitting accuracy is low ($> 10^{-2}$). For slightly higher accuracy ($< 10^{-2}$), least-squares methods always result in interpolation, producing no data reduction. For fitting with B-splines, interpolation produces a curve with more data than the original if the knots are counted. Various data reduction schemes [11, 12, 13] have been suggested by other authors. However it is very difficult to make any reduction with these methods when the required accuracy is high.

Kallay [7] presented a method to fit a G^1 polynomial curve to a planar set of points with tangent directions. Piegl [6] and Chou [8, 9] also devised methods to fit C^1 rational B-spline curves to the same data for planar and for three-space cases. However, none of these methods have achieved high data reduction rates. Besides, there are systems which are based on integral B-splines, hence, can not utilize the methods in [8, 9].

In this paper, we describe a method which follows the same divide-and-conquer strategy used in [6, 8, 9] but produces an integral B-spline fitting curve. A new tolerance checking method is presented to speed up the tolerance checking process. The data reduction rate is compared favorably with that of previous methods.

The method is as follows: first, find the longest run of data that can be fit with one cubic Bézier curve, starting from the last fit data point. At the beginning, the longest run starts from the first data point. This process repeats until all the data are fit. The Bézier curves from the fits are then connected together to form a G^1 B-spline curve.

To locate the longest run of data that can be fit by one cubic Bézier curve, an attempt is made to fit the given list of data. If the data can not be fit by one Bézier curve with the given tolerance, the set of data is reduced, and the process to fit one cubic Bézier curve is repeated recursively. If the data can be fit by one Bézier curve, the set of data is enlarged, and the process to fit one cubic Bézier curve is repeated recursively. The recursion stops when the the longest run of data that can be fit is found.

Given a list of data, the following steps are used to obtain the Bézier fitting curve:

1. The first and last control points of the Bézier curve are the first and last points of the data, respectively.
2. The tangents at the first and last points of the Bézier curve are in the tangent directions of the first and last points of the data, respectively.
3. A cubic Bézier interpolation curve is created for each of the inner data points (data except the first and last point).
4. A cubic Bézier fitting curve is created by weighting the interpolation curves.
5. Checks are performed to insure the fitting curve is within the given tolerance.

In Step 3, the interpolation curve has the same control points and tangent directions as the fitting curve. In addition, the interpolation curve goes through the inner data point. A planar case happens when all of the first and last points, the end tangent directions, the inner data point, and the tangent direction at the inner data point are on a plane. In this case, the tangent direction of the interpolation curve at the inner data point also agrees to the given tangent direction at the point.

In the following sections, Steps 3, 4 and 5 are discussed in detail.

2 Interpolation Cubic Computation

In this section we discussed the methods to obtain the interpolation curves. Most of the equations in this section can also be found in [8, 9].

A cubic Bézier curve can be written as:

$$C(u) = P_0(B_0^3(u) + B_1^3(u)) + \alpha t_0 B_1^3(u) + \beta t_1 B_2^3(u) + P_3(B_2^3(u) + B_3^3(u)) \quad (1)$$

where

- $B_i^3(u) = \binom{3}{i} u^i (1-u)^{3-i}$ are the cubic Bernstein polynomials;
- t_0 and t_1 are the end tangent directions (unit length); and
- P_0, P_3, $P_1 = P_0 + \alpha t_0$, $P_2 = P_3 + \beta t_1$ are the control points.

The above equation assumes, without loss of generality, that the cubic starts at $u = 0$ and ends at $u = 1$. An illustration of the variables is in Figure 1.

For a given data point P, the interpolation curve goes through this point.

$$P = P_0(B_0^3(\bar{u}) + B_1^3(\bar{u})) + \alpha t_0 B_1^3(\bar{u}) + \beta t_1 B_2^3(\bar{u}) + P_3(B_2^3(\bar{u}) + B_3^3(\bar{u})) \quad (2)$$

There are three variables in Equation (2): α, β, and $\bar{u}$. Equation (2) actually represents three equations, for the x, y, and z components, respectively. When t_0, t_1, and $\overline{P_0P_3}$ span the three-space, it is proved in [8] that there exists at most one set of solution $(\alpha, \beta, \bar{u})$ with $\bar{u} \in [0, 1]$. $\bar{u}$ can be found by performing dot products on Equation (2) with $t_3 = t_0 \times t_1$:

$$\frac{(P - P_0)t_3}{(P_3 - P_0)t_3} = B_2^3(\bar{u}) + B_3^3(\bar{u}) \quad (3)$$

This is a cubic Equation in $\bar{u}$. Exactly one solution exists in $[0,1]$, if and only if $\frac{(P-P_0)t_3}{(P_3-P_0)t_3} \in [0,1]$. Once $\bar{u}$ is obtained, α and β, and hence the interpolation curve, can be found from Equation (2).

When t_0, t_1, and $\overline{P_0P_3}$ are coplanar, Equation (2) has no solution if P is not in the plane. If P is in the plane, Equation (2) provides only two independent equations. If the tangent direction at P, denoted as t_2 (unit length), is also in the plane, an additional equation is provided by forcing the interpolation curve to agree to t_2 at P.

$$\delta t_2 = (P_3 - P_0)B_1^2(\bar{u}) + \alpha t_0(B_0^2(\bar{u}) - B_1^2(\bar{u})) + \beta t_1(B_1^2(\bar{u}) - B_2^2(\bar{u})), \tag{4}$$

where δ is a multiplier, and $B_i^2(\bar{u}) = \binom{2}{i}\bar{u}^i(1-\bar{u})^{2-i}$ are the quadratic Bernstein polynomials.

Equations (2) and (4) provide four equations and have unknowns: $(\alpha, \beta, \delta, \bar{u})$. The equations can be combined and simplified to form a cubic equation in $\bar{u}$ [9].

$$\begin{aligned} -[((P-P_0)\cdot\bar{t}_2)(t_1\cdot\bar{t}_0) + ((P-P_0)\cdot\bar{t}_0)(t_1\cdot\bar{t}_2)] - \\ 3[((P-P_0)\cdot\bar{t}_2)(t_0\cdot\bar{t}_1)]\bar{u} + \\ 3[((P_3-P_0)\cdot\bar{t}_1)(t_0\cdot\bar{t}_2)]\bar{u}^2 + \\ [((P_3-P_0)\cdot\bar{t}_1)(t_2\cdot\bar{t}_0) + ((P_3-P_0)\cdot\bar{t}_0)(t_2\cdot\bar{t}_1)]\bar{u}^3 = 0, \end{aligned} \tag{5}$$

where

$$\hat{k} \text{ is the normal of the plane}, \bar{t}_0 = \hat{k}\times t_0, \bar{t}_1 = \hat{k}\times t_1, \bar{t}_2 = \hat{k}\times t_2. \tag{6}$$

Once $\bar{u}$ is obtained, α, β, and δ can be computed. From α and β the interpolation Bézier curve can be obtained. However, in the planar case, there may be more than one solution set $(\alpha, \beta, \delta, \bar{u})$ with $\bar{u} \in [0,1]$. Even with the restrictions that the tangent directions of the Bézier curve should agree to the given data, i.e., $\alpha > 0, \beta < 0$, and $\delta > 0$, multiple solutions may still occur. In addition, Equations (2) and (4) do not have solutions in some special cases, for example, when t_0, t_1, and t_2 are parallel.

3 Fitting Curve Determination

When multiple solutions occur in computing the interpolation curve, the solution that has the closest matching end tangents with those from the neighboring data point is chosen as the interpolation curve. On the other hand, a valid interpolation curve may not exist. In such a case, the data point does not participate in determining the fitting curve (Step 4).

The fitting curve is obtained by weighting the interpolation curves. Since all the interpolation curves have the same start and end points and the same start and end tangent directions, locating the two inner control points (P_1 and P_2) is enough to fix the fitting curve. This is equivalent to determining the α and β of the fitting curve, denoted as $\bar{\alpha}$ and $\bar{\beta}$. We express $\bar{\alpha}$ as a weighted sum of αs from the interpolation curves, denoted as α_i.

$$\bar{\alpha} = \frac{\sum_i w_i \alpha_i}{\sum_i w_i} \tag{7}$$

Three possible methods to compute $\bar{\alpha}$ are listed below. $\bar{\beta}$ can be computed by similar methods.

1: $w_i = 1$.

2: $w_j = 1$, when $\alpha_j = \max$ or $\min \alpha_i$; $w_i = 0$, for the rest of i.

3: $w_i = B_1^3(\bar{u}_i)$, where $\bar{u}_i$ is the parameter of the interpolation curve at the ith data point.

Method 1 gives even weights to all the interpolation curves. Roughly speaking, Method 2 takes the average of the maximum and minimum interpolation curves. The third method gives heavier weight to the interpolation curve whose data point is closer to P_0. This results in a fitting curve that deviates less from the data point when the point is closer to P_0. Since P_1 has greater influence than P_2 on the part of the curve closer to P_0, Method 3 reduces the error of the fitting curve. Therefore, Method 3 is used in this paper. The formula $w_i = B_2^3(\bar{u}_i)$ is used to compute $\bar{\beta}$.

4 Tolerance Handling

Since we are interested in achieving high accuracy, we would like to check the distance between the data point and the fitting curve. However, it is quite expensive to compute the closest distance from a point to a curve. In order to reduce this cost, we first perform two pre-tests: one gives a quick rejection of the fitting curve, the other raises a quick acceptance of the fit. If the fit falls through both of the tests, the more costly closest distance computation is performed.

When we compute the interpolation curves for a list of data, we keep the maximum and minimum values of the $w_i\alpha_i$ (and $w_i\beta_i$) for the curves obtained so far. When the difference between the maximum and minimum is greater than a constant value, we declare the fit to be unsuccessful. This condition may be reached before all the data points are processed, and usually it happens during the first several recursive runs of fitting. Hence, this test serves as a quick rejection to data sets that variate too much to be fit by a cubic, and it significantly accelerates the process in locating the longest run of data. A good value for the constant that determines the rejection is two order of magnitude of the given tolerance for a tolerance of 10^{-4}. This formula is purely experiential. Note that this method may declare a fit to be unsuccessful, even though the fitting curve is within the given tolerance.

For the quick acceptance test, we observe that

$$|C-P| \leq |C(\bar{u})-P| = |\Delta\alpha B_1^3(\bar{u})+\Delta\beta B_2^3(\bar{u})| \leq |\Delta\alpha| B_1^3(\bar{u})+|\Delta\beta| B_2^3(\bar{u}), \quad (8)$$

where $\Delta\alpha = \alpha - \bar{\alpha}$ and $\Delta\beta = \beta - \bar{\beta}$. Both $B_1^3(\bar{u})$ and $B_2^3(\bar{u})$ are available from the process computing the interpolation curve. Therefore, for each point, this estimation of the upper bound for the error can be obtained by performing two multiplications, two subtractions and one addition.

Finally, computing the closest distance between the fitting curve and a data point is necessary only if the data point fails the quick acceptance test. When this happens, we use the Newton-Raphson method to locate the closest point on the curve to the data point. The parameter $\bar{u}$ of the interpolation curve at the data point serves as a very good initial guessing point for the Newton-Raphson method.

5 Examples

In this section, we show the results from applying the above fitting method to four data sets. The first data set (Half-Circle) is created by sampling 101 points on a half circle. The second data set (Torus-Plane) with 361 points is created by tracing the intersection curve of a torus and a plane. Both of these data sets are planar. But only the first data set is in the x-y plane. The third data set (Planar-Spiral) contains a planar section and a spiral three-space section with a total of 51 data points. The remaining data set (Mixed-Spiral) has 101 data points with mixed planar and three-space sections. Figures from the sample fits are shown in Figures 2 to 5. The first three figures can be compared with those from the rational curve fits in [9]. The last figure can be compared with that in [8].

	Half-Circle		Torus-Plane		Planar-Spiral			Mixed-Spiral	
Tolerance	10^{-3}	10^{-6}	10^{-2}	10^{-4}	$2 \cdot 10^{-1}$	10^{-3}	10^{-5}	10^{-3}	10^{-5}
From Input	(101) 303		(361) 1805		(51) 255			(101) 505	
Current Method	(10) 34	(22) 70	(34) 140	(61) 248	(13) 56	(34) 140	(40) 164	(64) 260	(85) 344
Method in [9]	(10) 44	(58) 236	(43) 219	(112) 564	(16) 84	(37) 189	(61) 309	(82) 414	(100) 504
Inter-polation	(152) 459		(602) 2411		(85) 344			(169) 678	

Table 1: Comparison of Data Storage in Number of Floating Point Data.

The amount of floating point data in the input data and the fitting curves is listed in Table 1. Every input data point is counted as five floating point data: three for its position and two for its tangent direction, except when the data is in the x-y plane, in which case every input data point is counted as three floating point data. For the fitting curves, we count three floating point data for each control point, except when the curve is in the x-y plane, in which case we count two data for each control point. Each knot in the fitting curves is also counted as one floating point data. Table 1 also compares the current method with the rational fitting method in [9]. In the table, the number in the parentheses is the number of control points in the fitting curve.

As mentioned earlier, for most of the tolerances given in Table 1, least-squares methods result in interpolation. Hence, the number of floating point data from interpolation is also listed in Table 1 for comparison.

Table 2 lists the order of magnitude of the maximum deviation of the tangent direction of the fitting curves. The deviation is measured at the data points. Table 2 also contains the acceptance ratio, the inverse of the ratio of the number of closest point computation versus the number of points accepted without the closest point computation. The larger this ratio, the more closest point computations have been saved by the quick acceptance tests.

	Half-Circle		Torus-Plane		Planar-Spiral			Mixed-Spiral	
Tolerance	10^{-3}	10^{-6}	10^{-2}	10^{-4}	$2 \cdot 10^{-1}$	10^{-3}	10^{-5}	10^{-3}	10^{-5}
Tangent Deviation	10^{-3}	10^{-6}	10^{-2}	10^{-3}	10^{-1}	10^{-3}	10^{-6}	10^{-3}	10^{-6}
Acceptance Ratio	.14	.01	.66	.03	1.27	1.67	.45	.95	2.42

Table 2: Tangent Deviation and Acceptance Ratio.

6 Discussion and Conclusion

From Table 1 we can see that the current method provides much better data reduction than the previous methods. The difference in the reduction ratio becomes wider when the tolerance is smaller. However, the current method only provides G^1 continuous curves. Although the fitting curve can be made C^1 by adjusting knot intervals, the resulting curve may have very bad parametrization. With the rational fitting method in [8, 9], the curve can be made C^1 by adjusting weights of the curve while maintaining a reasonable parametric flow. Even though interpolation methods produce curves with the most data, they produce C^2 continuous curves, which may be important in some applications.

Table 2 shows that the tangent direction of the fitting curves has roughly the same error as the position of the curves, even though we do not explicitly test the tangent direction against the given tolerance. From all the test cases, we see the effectiveness of the quick acceptance test is mixed. It seems to be less effective when the data is planar.

Although the method presented here demonstrates great improvement in data reduction compared with existing methods, it still produces curves that require large amount of storage, especially when the required accuracy is very high. For example, in the Mixed-Spiral case, when tolerance = 10^{-5}, 85 control points are in the fitting curve for the 101 data points. Also, the storage for the knots becomes large, 89 knots in this case.

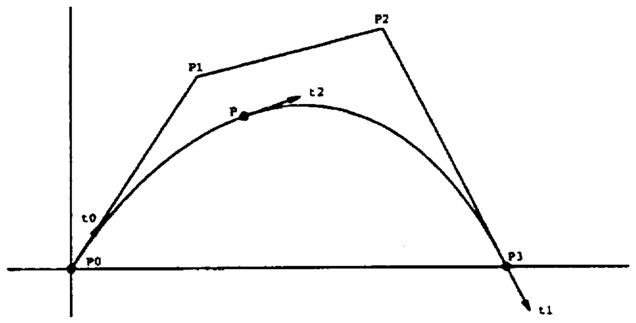

Figure 1: The given data points and tangent directions for an interpolation cubic.

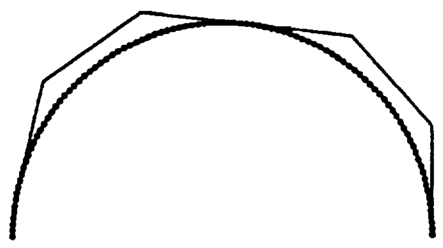

Figure 2: A nonrational fit of the data set Half-Circle with 10^{-3} tolerance.

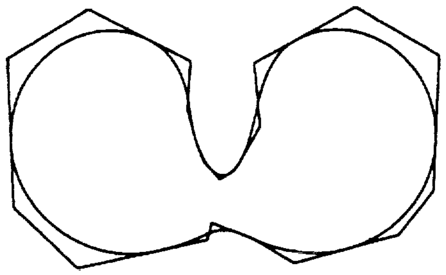

Figure 3: A nonrational fit of the data set Torus-Plane with 10^{-2} tolerance.

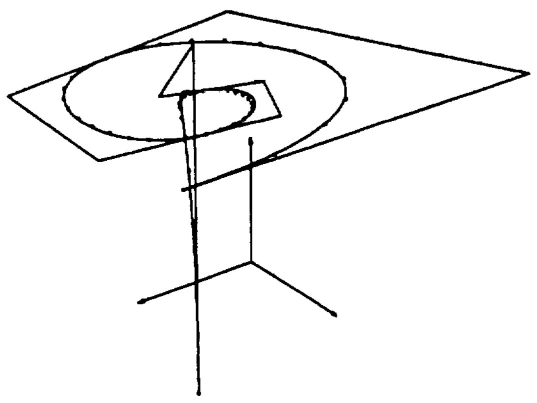

Figure 4: A nonrational fit of the data set Planar-Spiral with $2 \cdot 10^{-1}$ tolerance.

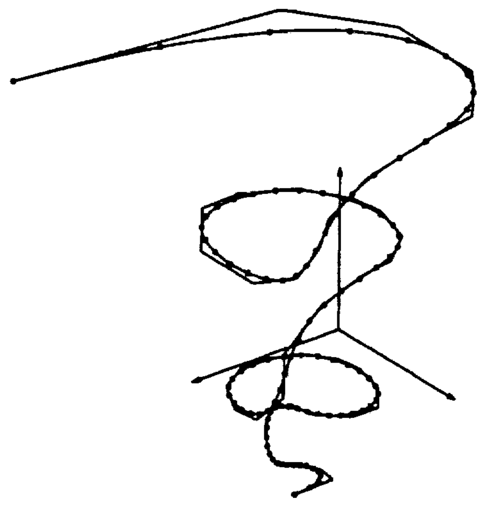

Figure 5: A nonrational fit of the data set Mixed-Spiral with 10^{-5} tolerance.

References

[1] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York (1978).

[2] W. Bohm, G. Farin, and J. Kahmann, A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design*, Vol. 1, No. 1, (1984) pp 1-60.

[3] G. Farin, *Curves and Surfaces For Computer Aided Geometric Design, A Practical Guide*, Academic Press, (1991).

[4] L. Piegl and W. Tiller, Curve and surface constructions using rational B-splines, *Computer-Aided Design*, Vol. 19, (1987) pp 485-498.

[5] L. Piegl, On NURBS: a survey, *IEEE Computer Graphics & Applications*, Vol. 11, (1991) pp 55-71.

[6] L. Piegl, A technique for smoothing scattered data with conic sections, *Computers in Industry*, Vol. 9, No. 3, Nov. 1987, pp 223-237.

[7] M. Kallay, Approximating a composite cubic curve with one fewer pieces, *Computer-Aided Design*, Vol. 19, (1987) pp 539-543.

[8] J. Chou and L. Piegl, Data Reduction Using Cubic Rational B-splines, *IEEE Computer Graphics & Applications*, Vol. 12, No. 3, (1992) pp 60-68.

[9] J. Chou and M. Blake, On planar cubics through a point at a direction, submitted to publication.

[10] D. Rogers, Constrained B-spline curve and surface fitting, *Computer-Aided Design*, Vol. 21, No. 10, Dec. 1989, pp. 641-648.

[11] U. Wever, Global and local data reduction strategies for cubic splines, *Computer-Aided Design*, Vol. 23, No. 2, Mar. 1991, pp 127-132.

[12] T Lyche, K. Morken, A data reduction strategy for splines, Research Report No 107, Institute of Informatics, University of Oslo, Feb. 1987.

[13] J. Hoschek, Approximate conversion of spline curves, *Computer-Aided Design*, Vol 4, 1987, pp 59-66.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE October 1993	3. REPORT TYPE AND DATES COVERED Contractor Report

4. TITLE AND SUBTITLE Achieving High Data Reduction with Integral Cubic B-Splines	5. FUNDING NUMBERS NAS2-12961
6. AUTHOR(S) Jin J. Chou	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Computer Sciences Corporation P.O. Box 390757 Mountain View, CA 94039	8. PERFORMING ORGANIZATION REPORT NUMBER A-94011
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-177628

11. SUPPLEMENTARY NOTES
Point of Contact: Matt Blake, Ames Research Center, MS T045-2, Moffett Field, CA 94035-1000
(415) 604-4978

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified — Unlimited Subject Category 67	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

During geometry processing, tangent directions at the data points are frequently readily available from the computation process that generates the points. It is desirable to utilize this information to improve the accuracy of curve fitting and to improve data reduction.

This paper presents a curve fitting method which utilizes both position and tangent direction data. This method produces G^1 nonrational B-spline curves. From the examples, the method demonstrates very good data reduction rates while maintaining high accuracy in both position and tangent direction.

14. SUBJECT TERMS Data reduction, Cubic B-splines, Curve fitting	15. NUMBER OF PAGES 16
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT